REJ10J1828-0100

Everywhere you imagine. RENESAS

SuperH™ Family E10A-USB Emulator

Additional Document for User's Manual

Supplementary Information on Using the SH7147 Group Debugging MCU Board

Renesas Microcomputer Development Environment System

SuperH™ Family

E10A-USB for SH7147 HS7147DBKCU01HE

User's Manual

Rev.1.00

Revision Date: Apr. 04, 2008

Renesas Technology
www.renesas.com

Notes regarding these materials

1. This document is provided for reference purposes only so that Renesas customers may select the appropriate Renesas products for their use. Renesas neither makes warranties or representations with respect to the accuracy or completeness of the information contained in this document nor grants any license to any intellectual property rights or any other rights of Renesas or any third party with respect to the information in this document.
2. Renesas shall have no liability for damages or infringement of any intellectual property or other rights arising out of the use of any information in this document, including, but not limited to, product data, diagrams, charts, programs, algorithms, and application circuit examples.
3. You should not use the products or the technology described in this document for the purpose of military applications such as the development of weapons of mass destruction or for the purpose of any other military use. When exporting the products or technology described herein, you should follow the applicable export control laws and regulations, and procedures required by such laws and regulations.
4. All information included in this document such as product data, diagrams, charts, programs, algorithms, and application circuit examples, is current as of the date this document is issued. Such information, however, is subject to change without any prior notice. Before purchasing or using any Renesas products listed in this document, please confirm the latest product information with a Renesas sales office. Also, please pay regular and careful attention to additional and different information to be disclosed by Renesas such as that disclosed through our website. (http://www.renesas.com)
5. Renesas has used reasonable care in compiling the information included in this document, but Renesas assumes no liability whatsoever for any damages incurred as a result of errors or omissions in the information included in this document.
6. When using or otherwise relying on the information in this document, you should evaluate the information in light of the total system before deciding about the applicability of such information to the intended application. Renesas makes no representations, warranties or guaranties regarding the suitability of its products for any particular application and specifically disclaims any liability arising out of the application and use of the information in this document or Renesas products.
7. With the exception of products specified by Renesas as suitable for automobile applications, Renesas products are not designed, manufactured or tested for applications or otherwise in systems the failure or malfunction of which may cause a direct threat to human life or create a risk of human injury or which require especially high quality and reliability such as safety systems, or equipment or systems for transportation and traffic, healthcare, combustion control, aerospace and aeronautics, nuclear power, or undersea communication transmission. If you are considering the use of our products for such purposes, please contact a Renesas sales office beforehand. Renesas shall have no liability for damages arising out of the uses set forth above.
8. Notwithstanding the preceding paragraph, you should not use Renesas products for the purposes listed below:
   (1) artificial life support devices or systems
   (2) surgical implantations
   (3) healthcare intervention (e.g., excision, administration of medication, etc.)
   (4) any other purposes that pose a direct threat to human life

   Renesas shall have no liability for damages arising out of the uses set forth in the above and purchasers who elect to use Renesas products in any of the foregoing applications shall indemnify and hold harmless Renesas Technology Corp., its affiliated companies and their officers, directors, and employees against any and all damages arising out of such applications.
9. You should use the products described herein within the range specified by Renesas, especially with respect to the maximum rating, operating supply voltage range, movement power voltage range, heat radiation characteristics, installation and other product characteristics. Renesas shall have no liability for malfunctions or damages arising out of the use of Renesas products beyond such specified ranges.
10. Although Renesas endeavors to improve the quality and reliability of its products, IC products have specific characteristics such as the occurrence of failure at a certain rate and malfunctions under certain use conditions. Please be sure to implement safety measures to guard against the possibility of physical injury, and injury or damage caused by fire in the event of the failure of a Renesas product, such as safety design for hardware and software including but not limited to redundancy, fire control and malfunction prevention, appropriate treatment for aging degradation or any other applicable measures. Among others, since the evaluation of microcomputer software alone is very difficult, please evaluate the safety of the final products or system manufactured by you.
11. In case Renesas products listed in this document are detached from the products to which the Renesas products are attached or affixed, the risk of accident such as swallowing by infants and small children is very high. You should implement safety measures so that Renesas products may not be easily detached from your products. Renesas shall have no liability for damages arising out of such detachment.
12. This document may not be reproduced or duplicated, in any form, in whole or in part, without prior written approval from Renesas.
13. Please contact a Renesas sales office if you have any questions regarding the information contained in this document, Renesas semiconductor products, or if you have any other inquiries.

Contents

Section 1 About the Emulator 1

1.1 Components of the Emulator 1

1.2 Connecting the Emulator with the SH7147 Group Debugging MCU Board 3

Section 2 Software Specifications when Using the SH7147 Group Debugging MCU Board 5

2.1 Differences between the MCU and the Emulator 5

2.2 Specific Functions and Notes for the Emulator when Using the SH7147 Group Debugging MCU Board 8

2.2.1 Selecting the Target MCU 8

2.2.2 Event Condition Functions 8

2.2.3 Trace Functions 14

2.2.4 Note on Using the JTAG (H-UDI) Clock (TCK) 25

2.2.5 Notes on Setting the [Breakpoint] Dialog Box 25

2.2.6 Notes on Setting the [Event Condition] Dialog Box and the BREAKCONDITION_SET Command 25

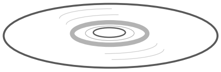
2.2.7 Performance Measurement Function 26

Section 1 About the Emulator

1.1 Components of the Emulator

The E10A-USB emulator supports the SH7147 group debugging MCU board. Table 1.1 lists the components of the emulator.

Table 1.1 Components of the Emulator

Classification	Component	Appearance	Quantity	Remarks
Hardware	Emulator box		1	HS0005KCU01H: Depth: 65.0 mm, Width: 97.0 mm, Height: 20.0 mm, Mass: 72.9 g or HS0005KCU02H: Depth: 65.0 mm, Width: 97.0 mm, Height: 20.0 mm, Mass: 73.7 g
	User system interface cable		1	14-pin type: Length: 20 cm, Mass: 33.1 g
	User system interface cable		1	36-pin type: Length: 20 cm, Mass: 49.2 g (only for HS0005KCU02H)
	USB cable		1	Length: 150 cm, Mass: 50.6 g
Software	E10A-USB emulator setup program, SuperH™ Family E10A-USB Emulator User's Manual, Supplementary Information on Using the SH7147 Group Debugging MCU Board*, and Test program manual for HS0005KCU01H and HS0005KCU02H		1	HS0005KCU01SR, HS0005KCU01HJ, HS0005KCU01HE, HS7147DBKCU01HJ, HS7147DBKCU01HE, HS0005TM01HJ, and HS0005TM01HE (provided on a CD-R)

Note: Additional document for the MCUs supported by the emulator is included. Check the target MCU and refer to its additional document.

1.2 Connecting the Emulator with the SH7147 Group Debugging MCU Board

Table 1.2 shows the type number of the emulator, the corresponding connector type, and the use of AUD function.

Table 1.2 Type Number, AUD Function, and Connector Type

Type Number	Connector	AUD Function
HS0005KCU02H	36-pin connector	Available
HS0005KCU01H, HS0005KCU02H	14-pin connector	Not available

The H-UDI port connector has the 36-pin and 14-pin types as described below. Use them according to the purpose of the usage.

1. 36-pin type (with AUD function)
   The AUD trace function is supported. A large amount of trace information can be acquired in realtime. The window trace function is also supported for acquiring memory access in the specified range (memory access address or memory access data) by tracing.
2. 14-pin type (without AUD function)
   The AUD trace function cannot be used because only the H-UDI function is supported.

Section 2 Software Specifications when Using the SH7147 Group Debugging MCU Board

2.1 Differences between the MCU and the Emulator

1. When the emulator system is initiated, it initializes the general registers and part of the control registers. The initial values of the MCU are undefined. When the emulator is initiated from the workspace, a value to be entered is saved in a session.

Table 2.1 Register Initial Values at Emulator Link Up

Register	Emulator at Link Up
R0 to R14	H'00000000
R15 (SP)	Value of the SP in the power-on reset vector table
PC	Value of the PC in the power-on reset vector table
SR	H'000000F0
GBR	H'00000000
VBR	H'00000000
MACH	H'00000000
MACL	H'00000000
PR	H'00000000

2. The emulator uses the H-UDI; do not access the H-UDI.
3. Low-Power States
   - When the emulator is used, the sleep state can be cleared with either the clearing function or with the [STOP] button, and a break will occur.
   - The memory must not be accessed or modified in software standby state.
   - When the emulator is used, do not use the deep software standby mode.

4. Reset Signals

   The MCU reset signals are only valid during emulation started with clicking the GO or STEP-type button. If these signals are enabled on the user system in command input wait state, they are not sent to the MCU.

Note: Do not break the user program when the _RES, _BREQ, or _WAIT signal is being low. A TIMEOUT error will occur. If the _BREQ or _WAIT signal is fixed to low during break, a TIMEOUT will occur at memory access. (Some MCUs will incorporate no _BREQ or _WAIT.)

5. Data Transfer Controller (DTC)

   When the MCU incorporates a DTC, the DTC operates even when the emulator is used. When a data transfer request is generated, the DMAC executes DTC transfer.

6. Memory Access during User Program Execution

   During execution of the user program, memory is accessed by the following two methods, as shown in table 2.2.

Table 2.2 Memory Access during User Program Execution

Method	Description
H-UDI read/write	The stopping time of the user program is short because memory is accessed by the dedicated bus master.
Short break	This method is not used in this product. (Do not set short break.)

The method for accessing memory during execution of the user program is specified by using the [Configuration] dialog box.

Table 2.3 Stopping Time by Memory Access (Reference)

Method	Condition	Stopping Time
H-UDI read/write	Reading of one longword for the internal RAM	Reading: Maximum 2 bus clocks (Bϕ)
	Writing of one longword for the internal RAM	Writing: Maximum 2 bus clocks (Bϕ)

7. Memory Access to the External Flash Memory Area

   The emulator can download the load module to the external flash memory area (for details, refer to section 6.22, Download Function to the Flash Memory Area, in the SH Family E10A-USB Emulator User′s Manual). Neither memory write nor BREAKPOINT setting is enabled for the external flash memory area. To set the break condition for the program on the external flash memory, use the Event Condition function. Some MCUs will incorporate no external flash memory area.

8. Using WDT

   The WDT does not operate during break.

9. Loading Sessions

   Information in [JTAG clock] of the [Configuration] dialog box cannot be recovered by loading sessions. Thus the TCK value will be as follows:

   — When HS0005KCU01H or HS0005KCU02H is used: TCK = 2.5 MHz

10. [IO] Window

   — Display and modification

   For each watchdog timer register, there are two registers to be separately used for write and read operations.

Table 2.4 Watchdog Timer Register

Register Name	Usage	Register
WTCSR (W)	Write	Watchdog timer control/status register
WTCNT (W)	Write	Watchdog timer counter
WTCSR(R)	Read	Watchdog timer control/status register
WTCNT(R)	Read	Watchdog timer counter

   — Customization of the I/O-register definition file

   After the I/O-register definition file is created, the MCU’s specifications may be changed. If each I/O register in the I/O-register definition file differs from addresses described in the hardware manual, change the I/O-register definition file according to the description in the hardware manual. The I/O-register definition file can be customized depending on its format. Note that, however, the emulator does not support the bit-field function.

   — Verify

   In the [IO] window, the verify function of the input value is disabled.

9. Illegal Instructions

   Do not execute illegal instructions with STEP-type commands.

10. MCU Operating Mode

   Do not use the emulator in the boot mode, user boot mode, and user program mode.

11. MCU On-chip Flash Memory

   The MCU on-chip flash memory cannot be reprogrammed during execution of the user program.

2.2 Specific Functions and Notes for the Emulator when Using the SH7147 Group Debugging MCU Board

2.2.1 Selecting the Target MCU

Select the target MCU for the emulator according to that in use. For details, refer to section 4, Preparations for Debugging, in the SuperH™ Family E10A-USB Emulator User's Manual.

For debugging the SH7147 Group debugging MCU board, select the following in the [Debugger Target] and [Select Emulator mode] dialog boxes.

- [Debugger Target] dialog box
  — SH7147 Group Debug MCU BOARD E10A-USB SYSTEM (CPU SH-2)
- [Select Emulator mode] dialog box
  — For R5F71474: R5F71474_Debug_MCU_BOARD
  — For R5F71475: R5F71475_Debug_MCU_BOARD
  — For R5F 71476: R5F71476_Debug_MCU_BOARD
  — For R5F 71424: R5F71424_Debug_MCU_BOARD
  — For R5F 71426: R5F71426_Debug_MCU_BOARD

2.2.2 Event Condition Functions

The emulator is used to set event conditions for the following three functions:

- Break of the user program
- Internal trace
- Start or end of performance measurement

Table 2.5 lists the types of Event Condition.

Table 2.5 Types of Event Condition

Event Condition Type	Description
Address bus condition (Address)	Sets a condition when the address bus (data access) value or the program counter value (before or after execution of instructions) is matched.
Data bus condition (Data)	Sets a condition when the data bus value is matched. Byte, word, or longword can be specified as the access data size.
Bus state condition (Bus State)	There are two bus state condition settings: Bus state condition: Sets a condition when the data bus value is matched. Read/Write condition: Sets a condition when the read/write condition is matched.
Count	Sets a condition when the specified other conditions are satisfied for the specified counts.
Action	Selects the operation when a condition (such as a break, a trace halt condition, or a trace acquisition condition) is matched.

Using the [Combination action (Sequential or PtoP)] dialog box, which is opened by selecting [Combination action (Sequential or PtoP)] from the pop-up menu on the [Event Condition] sheet, specifies the sequential condition and the start or end of performance measurement.

Table 2.6 lists the combinations of conditions that can be set under Ch1 to Ch10.

Table 2.6 Dialog Boxes for Setting Event Conditions

		Function				
Dialog Box		Address Bus Condition (Address)	Data Bus Condition (Data)	Bus State Condition (Bus State)	Count Condition (Count)	Action
[Event Condition 1]	Ch1	O	O	O	O	O (B, T1, and P)
[Event Condition 2]	Ch2	O	O	O	X	O (B, T1, and P)
[Event Condition 3]	Ch3	O	X	X	X	O (B and T2)
[Event Condition 4]	Ch4	O	X	X	X	O (B and T3)
[Event Condition 5]	Ch5	O	X	X	X	O (B and T3)
[Event Condition 6]	Ch6	O	X	X	X	O (B and T2)
[Event Condition 7]	Ch7	O	X	X	X	O (B and T2)
[Event Condition 8]	Ch8	O	X	X	X	O (B and T2)
[Event Condition 9]	Ch9	O	X	X	X	O (B and T2)
[Event Condition 10]	Ch10	O	X	X	X	O (B and T2)

Notes: 1. O: Can be set in the dialog box.
X: Cannot be set in the dialog box.

2. For the Action item,
B: Setting a break is enabled. (For the count condition, setting a break is only enabled.)
T1: Setting the trace halt and acquisition conditions are enabled for the internal trace.
T2: Setting the trace halt is enabled for the internal trace.
T3: Setting the trace halt and point-to-point is enabled for the internal trace.
P: Setting a performance-measurement start or end condition is enabled.

Sequential Setting: Using the [Combination action (Sequential or PtoP)] dialog box specifies the sequential condition and the start or end of performance measurement.

Table 2.7 Conditions to Be Set

Classification	Item	Description
[Ch1, 2, 3] list box	Sets the sequential condition and the start or end of performance measurement using Event Conditions 1 to 3.	
	Don't care	Sets no sequential condition or the start or end of performance measurement.
	Break: Ch3-2-1	Breaks when a condition is satisfied in the order of Event Condition 3, 2, 1.
	Break: Ch2-1	Breaks when a condition is satisfied in the order of Event Condition 2, 1.
	I-Trace stop: Ch3-2-1	Halts acquisition of an internal trace when a condition is satisfied in the order of Event Condition 3, 2, 1.
	I-Trace stop: Ch2-1	Halts acquisition of an internal trace when a condition is satisfied in the order of Event Condition 2, 1.
	Ch2 to Ch1 PA	Sets the performance measurement period during the time from the satisfaction of the condition set in Event Condition 2 (start condition) to the satisfaction of the condition set in Event Condition 1 (end condition).
	Ch1 to Ch2 PA	Sets the performance measurement period during the time from the satisfaction of the condition set in Event Condition 1 (start condition) to the satisfaction of the condition set in Event Condition 2 (end condition).
[Ch4, 5] list box	Sets the point-to-point of the internal trace (the start or end condition of trace acquisition) using Event Conditions 4 and 5.	
	Don't care	Sets no start or end condition of trace acquisition.
	I-Trace: Ch5 to Ch4 PtoP	Sets the acquisition period during the time from the satisfaction of the condition set in Event Condition 5 (start condition) to the satisfaction of the condition set in Event Condition 4 (end condition).

Notes: 1. If the start condition is satisfied after the end condition has been satisfied by measuring performance, performance measurement will be restarted. For the measurement result after a break, the measurement results during performance measurement are added.

2. If the start condition is satisfied after the end condition has been satisfied by the point-to-point of the internal trace, trace acquisition will be restarted.
3. When the start or end of performance measurement is used, the count for specifying the condition of Event Condition 1 must be once.

Usage Example of Sequential Break Extension Setting: A tutorial program provided for the product is used as an example. For the tutorial program, refer to section 6, Tutorial, in the SuperH™ Family E10A-USB Emulator User's Manual.

The conditions of Event Condition are set as follows:

1. Ch3

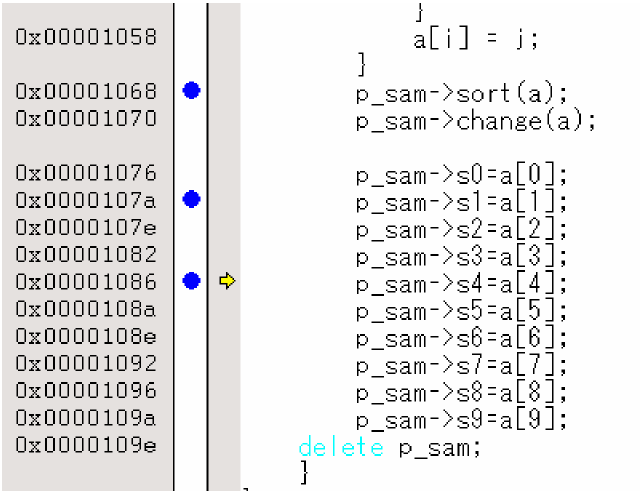
   Breaks address H'00001068 when the condition [Only program fetched address after] is satisfied.
2. Ch2

   Breaks address H'0000107a when the condition [Only program fetched address after] is satisfied.
3. Ch1

   Breaks address H'00001086 when the condition [Only program fetched address after] is satisfied.
   Note: Do not set other channels.
4. Sets the content of the [Ch1,2,3] list box to [Break: Ch 3-2-1] in the [Combination action (Sequential or PtoP)] dialog box.
5. Enables the condition of Event Condition 1 from the popup menu by clicking the right mouse button on the [Event Condition] sheet.

Then, set the program counter and stack pointer (PC = H'00000800, R15 = H'00010000) in the [Registers] window and click the [Go] button. If this does not execute normally, issue a reset and execute the above procedures.

The program is executed up to the condition of Ch1 and halted. Here, the condition is satisfied in the order of Ch3 -> 2 -> 1.



```
                                 }
0x00001058                       a[i] = j;
                             }
0x00001068   ●               p_sam->sort(a);
0x00001070                   p_sam->change(a);

0x00001076                   p_sam->s0=a[0];
0x0000107a   ●               p_sam->s1=a[1];
0x0000107e                   p_sam->s2=a[2];
0x00001082                   p_sam->s3=a[3];
0x00001086   ●   ⇨           p_sam->s4=a[4];
0x0000108a                   p_sam->s5=a[5];
0x0000108e                   p_sam->s6=a[6];
0x00001092                   p_sam->s7=a[7];
0x00001096                   p_sam->s8=a[8];
0x0000109a                   p_sam->s9=a[9];
0x0000109e               delete p_sam;
                         }
```

Figure 2.1 [Source] Window at Execution Halted (Sequential Break)

If the sequential condition, performance measurement start/end, or point-to-point for the internal trace is set, conditions of Event Condition to be used will be disabled. Such conditions must be enabled from the popup menu by clicking the right mouse button on the [Event Condition] sheet.

Notes: 1. If the Event condition is set for the slot in the delayed branch instruction by the program counter (after execution of the instruction), the condition is satisfied before executing the instruction in the branch destination (when a break has been set, it occurs before executing the instruction in the branch destination).

2. Do not set the Event condition for the SLEEP instruction by the program counter (after execution of the instruction).
Do not set the data access condition before executing one or two instructions in the SLEEP instruction.

3. If the power-on reset and the Event condition are matched simultaneously, no condition will be satisfied.

4. If a condition of which intervals are satisfied closely is set, no sequential condition will be satisfied. Set the Event conditions sequentially, which are satisfied closely, by the program counter with intervals of two or more instructions.
The CPU is structured as a pipeline; the order between the instruction fetch cycle and the memory cycle is determined by the pipeline. Accordingly, when the channel condition is matched in the order of bys cycle, the sequential condition is satisfied.

5. If the settings of the Event condition or the sequential conditions are changed during execution of the program, execution will be suspended. (The number of clocks to be suspended during execution of the program is a maximum of about 52 bus clocks (Bϕ). If the bus clock (Bϕ) is 10.0 MHz, the program will be suspended for 5.2 µs.)
6. If the settings of Event conditions or the sequential conditions are changed during execution of the program, the emulator temporarily disables all Event conditions to change the settings. During this period, no Event condition will be satisfied.
7. If the satisfaction is contended between the DTC transfer and conditions of Event Condition including the external bus access condition, the followings may be disabled: generation of a break after the satisfaction of conditions of Event Condition, halting and acquisition of the internal trace, and the start or end of performance measurement.
8. When the emulator is being connected, the user break controller (UBC) function is not available.

2.2.3 Trace Functions

The emulator supports the trace functions listed in table 2.8.

Table 2.8 Trace Functions

Function	Internal Trace	AUD Trace
Branch trace	Supported	Supported
Memory access trace	Supported	Supported
Software trace	Not supported	Supported

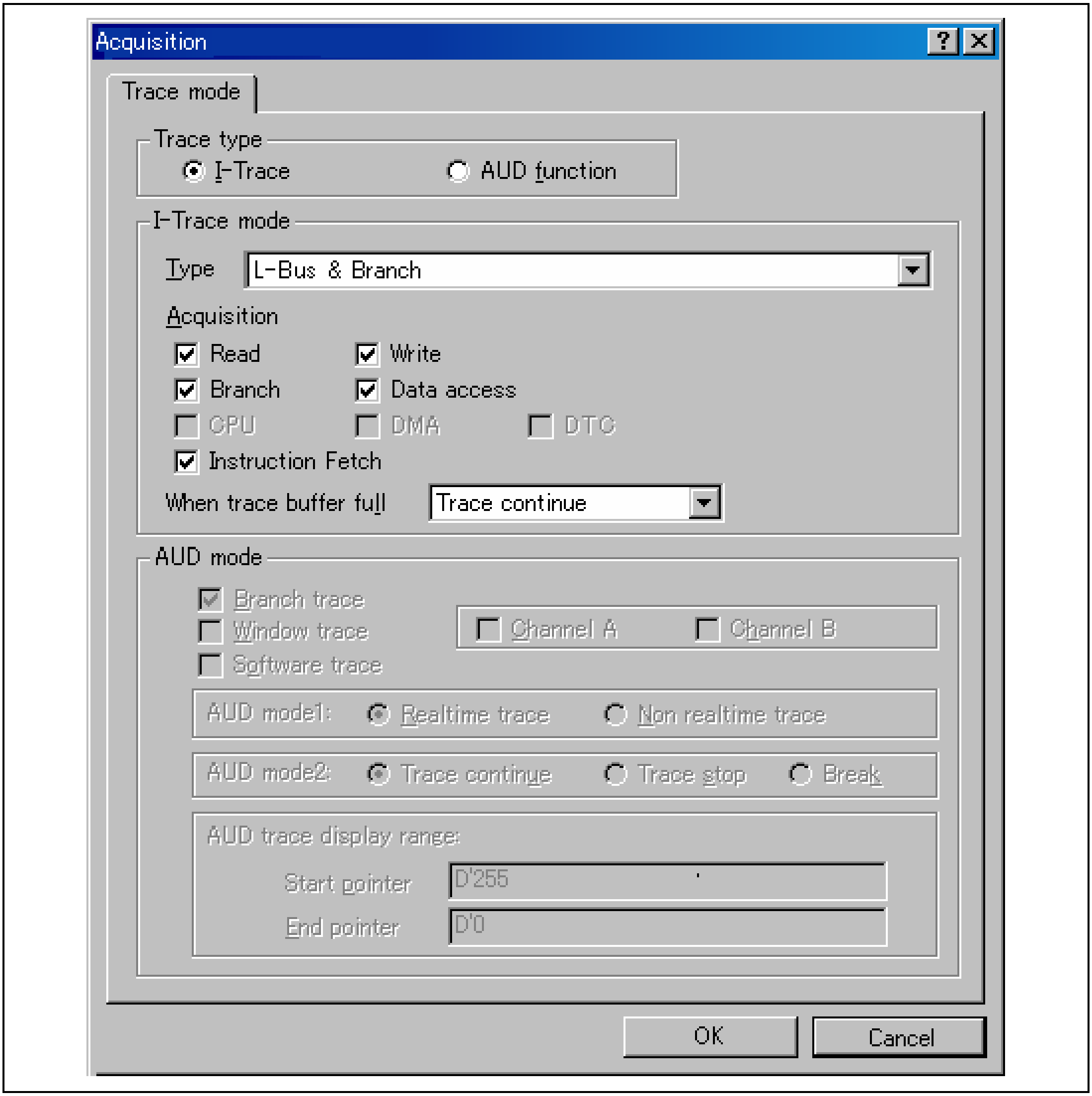
Table 2.9 shows the type numbers that the AUD function can be used.

Table 2.9 Type Number and AUD Function

Type Number	AUD Function
HS0005KCU01H	Not supported
HS0005KCU02H	Supported

The internal and AUD traces are set in the [Acquisition] dialog box of the [Trace] window.

Internal Trace Function: When [I-Trace] is selected for [Trace type] on the [Trace Mode] page of the [Acquisition] dialog box, the internal trace can be used.



Acquisition

Trace mode

Trace type
● I-Trace ○ AUD function

I-Trace mode
Type: L-Bus & Branch

Acquisition
☑ Read ☑ Write
☑ Branch ☑ Data access
□ CPU □ DMA □ DTC
☑ Instruction Fetch

When trace buffer full: Trace continue

AUD mode
☑ Branch trace
□ Window trace
□ Software trace
□ Channel A □ Channel B

AUD mode1: ● Realtime trace ○ Non realtime trace

AUD mode2: ● Trace continue ○ Trace stop ○ Break

AUD trace display range:
Start pointer: D'255
End pointer: D'0

Figure 2.2 [Acquisition] Dialog Box (Internal Trace Function)

The following three items can be selected as the internal trace from [Type] of [I-Trace mode].

Table 2.10 Information on Acquiring the Internal Trace

Item	Acquisition Information
[L-Bus & Branch]	Acquires the data and branch information on the L-bus. • Data access (read/write) • Branch information • Instruction fetch
[I-Bus]	Acquires the data on the I-bus. • Data access (read/write) • Selection of the bus master on the I-bus (CPU/DTC) • Instruction fetch
[I-Bus, L-Bus & Branch]	Acquires the contents of [L-Bus & Branch] and [I-Bus].

After selecting [Type] of [I-Trace mode], select the content to be acquired from [Acquisition]. Typical examples are described below (note that items disabled for [Acquisition] are not acquired).

For the MCUs that incorporate neither DMAC or DTC, do not select DMA or DTC when selecting the bus master on the I-bus.

- Example of acquiring branch information only:

  Select [L-Bus & Branch] from [Type] and enable [Branch] on [Acquisition].

- Example of acquiring the read or write access (L-bus) only by a user program:

  Select [L-Bus & Branch] from [Type] and enable [Read], [Write], and [Data access] on [Acquisition].

- Example of acquiring the read access only by DTC (I-bus):

  Select [I-Bus] from [Type] and enable [Read], [DTC], and [Data access] on [Acquisition].

Using Event Condition restricts the condition; the following three items are set as the internal trace conditions.

Table 2.11 Trace Conditions of the Internal Trace

Item	Acquisition Information
Trace halt	Acquires the internal trace until the Event Condition is satisfied. (The trace content is displayed in the [Trace] window after a trace has been halted. No break occurs in the user program.)
Trace acquisition condition	Acquires only the data access where the Event Condition is satisfied.
Point-to-point	Traces the period from the satisfaction of Event Condition 5 to the satisfaction of Event Condition 4.

To restrict trace acquisition to access for only a specific address or specific function of a program, an Event Condition can be used. Typical examples are described below.

- Example of halting a trace with a write access (L-bus) to H’FFFF8000 by the user program as a condition (trace halt):

  Set the condition to be acquired on [I-Trace mode].

  Set the following in the [Event Condition 1] or [Event Condition 2] dialog box:

  Address condition: Set [Address] and H’FFFF8000.

  Bus state condition: Set [L-Bus] and [Write].

  Action condition: Disable [Acquire Break] and set [Acquire Trace] for [Stop].

- Example of acquiring the write access (L-bus) only to H’FFFF8000 by the user program (trace acquisition condition):

  Select [L-Bus & Branch] from [Type] and enable [Write] and [Data access] on [Acquisition].

  Set the following in the [Event Condition 1] or [Event Condition 2] dialog box:

  Address condition: Set [Address] and H’FFFF8000.

  Bus state condition: Set [L-Bus] and [Write].

  Action condition: Disable [Acquire Break] and set [Acquire Trace] for [Condition].

For the trace acquisition condition, the condition to be acquired by Event Condition should be acquired by [I-Trace mode].

- Example of acquiring a trace for the period while the program passes H’1000 through H’2000 (point-to-point):

  Set the condition to be acquired on [I-Trace mode].

  Set the address condition as H’1000 in the [Event Condition 5] dialog box.

Set the address condition as H’2000 in the [Event Condition 4] dialog box.

Set [Ch4,5] as [I-Trace Ch5 to Ch4 PtoP] in the [Combination action (Sequential or PtoP)] dialog box.

When point-to-point and trace acquisition condition are set simultaneously, they are ANDed.

Notes on Internal Trace:

- Timestamp

  The timestamp is twice the crystal oscillator or the external clock that is connected to or input to the target MCU. Table 2.12 shows the timing for acquiring the timestamp.

Table 2.12 Timing for the Timestamp Acquisition

Item	Counter Value Stored in the Trace Memory
L-bus instruction fetch	Counter value when instruction fetch has been completed
L-bus data access	Counter value when data access has been completed
Branch	Counter value when the next bus cycle has been completed after a branch
I-bus fetch	Counter value when a fetch has been completed
I-bus data access	Counter value when data access has been completed

- Point-to-point

  The trace-start condition is satisfied when the specified instruction has been fetched. Accordingly, if the trace-start condition has been set for the overrun-fetched instruction (an instruction that is not executed although it has been fetched at a branch or transition to an interrupt), tracing is started during overrun-fetching of the instruction. However, when overrun-fetching is achieved (a branch is completed), tracing is automatically suspended.
  If the start and end conditions are satisfied closely, trace information will not be acquired correctly.
  The execution cycle of the instruction fetched before the start condition is satisfied may be traced.

- Halting a trace

  Do not set the trace end condition for the sleep instruction and the branch instruction that the delay slot becomes the sleep instruction.

- Trace acquisition condition

  Do not set the trace end condition for the sleep instruction and the branch instruction according to which the delay slot becomes the sleep instruction.
  When [I-Bus, L-Bus & Branch] is selected and the trace acquisition condition is set for the L-bus and I-bus with Event Condition, set the L-bus condition and the I-bus condition for [Event Condition 1] and [Event Condition 2], respectively.
  If the settings of [I-Trace mode] are changed during execution of the program, execution will be suspended. (The number of clocks to be suspended during execution of the program is a maximum of about 26 bus clocks (Bϕ). If the bus clock (Bϕ) is 10.0 MHz, the program will be suspended for 2.6 µs.)
  Do not use the data condition as the trace acquisition condition.

- Displaying a trace

  If a trace is displayed during execution of the program, execution will be suspended to acquire the trace information. (The number of clocks to be suspended during execution of the program is a maximum of about 16384 peripheral clocks (Pϕ) + 12310 bus clocks (Bϕ). If the peripheral clock (Pϕ) is 10.0 MHz and the bus clock (Bϕ) is 10.0 MHz, the program will be suspended for 2.87 ms.)
  If a break occurs with the Event Condition, when one or two instructions have been executed after a break occurred in an instruction and there is an unconditional branch, a trace result will be displayed even if the unconditional branch has not been executed. The trace acquisition result of the DTC transfer may not be displayed correctly. In such a case, a master where the trace has been generated or a line for displaying the trace will be blank.

- Restarting trace acquisition after halting

  Restarting trace acquisition is disabled during execution of the user program; a break must be generated.

- Note on execution of the user program

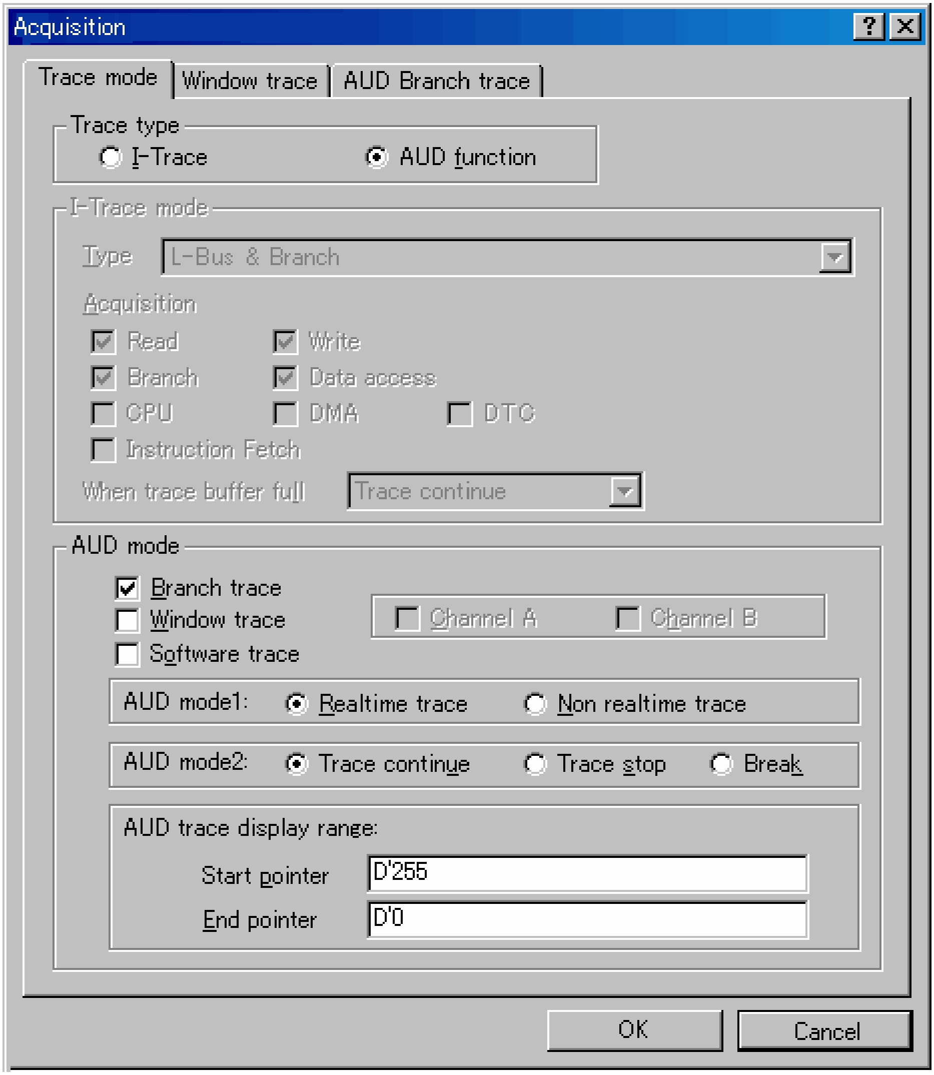
  Do not change trace settings during execution of the user program; trace acquisition may be disabled. The following trace settings will be changed: the conditions of Event Condition, the sequential conditions satisfied by Event Condition, and internal trace set in the [Acquisition] dialog box. To change these settings, a break must be generated.

AUD Trace Functions: This function is operational when the AUD pin of the device is connected to the emulator. Table 2.13 shows the AUD trace acquisition mode that can be set in each trace function.

Table 2.13 AUD Trace Acquisition Mode

Type	Mode	Description
Continuous trace occurs	Realtime trace	When the next branch occurs while the trace information is being output, all the information may not be output. The user program can be executed in realtime, but some trace information will be lost.
	Non realtime trace	When the next branch occurs while the trace information is being output, the CPU stops operations until the information is output. The user program is not executed in realtime.
Trace buffer full	Trace continue	This function overwrites the latest trace information to store the oldest trace information.
	Trace stop	After the trace buffer becomes full, the trace information is no longer acquired. The user program is continuously executed.

To set the AUD trace acquisition mode, click the [Trace] window with the right mouse button and select [Setting] from the pop-up menu to display the [Acquisition] dialog box. The AUD trace acquisition mode can be set in the [AUD mode1] or [AUD mode2] group box in the [Trace mode] page of the [Acquisition] dialog box.



Trace type
○ I-Trace ● AUD function

I-Trace mode

Type: L-Bus & Branch

Acquisition

☑ Read ☑ Write

☑ Branch ☑ Data access

□ CPU □ DMA □ DTC

□ Instruction Fetch

When trace buffer full: Trace continue

AUD mode

☑ Branch trace

□ Window trace

□ Software trace

□ Channel A □ Channel B

AUD mode1: ● Realtime trace ○ Non realtime trace

AUD mode2: ● Trace continue ○ Trace stop ○ Break

AUD trace display range:

Start pointer: D'255

End pointer: D'0

Figure 2.3 [Trace mode] Page

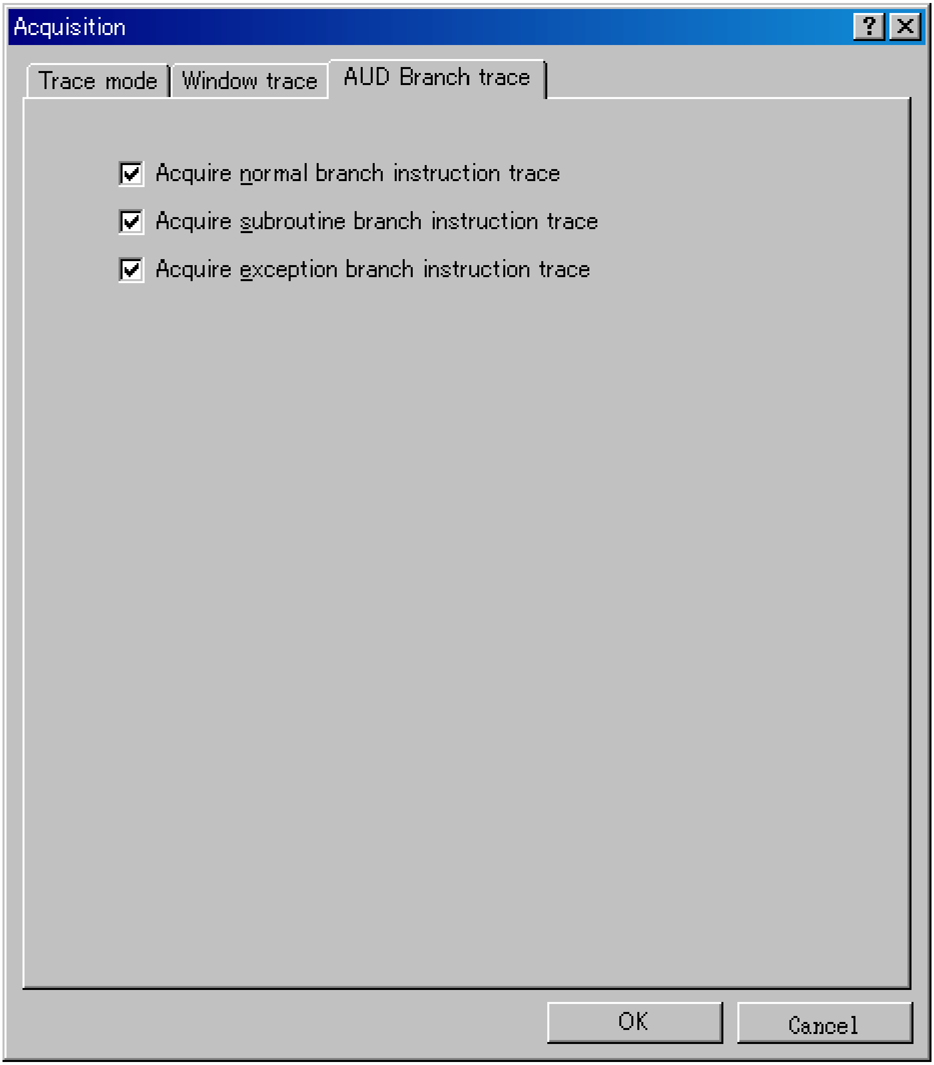
When the AUD trace function is used, select the [AUD function] radio button in the [Trace type] group box of the [Trace mode] page.

(a) Branch Trace Function

The branch source and destination addresses and their source lines are displayed.

Branch trace can be acquired by selecting the [Branch trace] check box in the [AUD function] group box of the [Trace mode] page.

The branch type can be selected in the [AUD Branch trace] page.



☑ Acquire normal branch instruction trace

☑ Acquire subroutine branch instruction trace

☑ Acquire exception branch instruction trace

Figure 2.4 [AUD Branch trace] Page

(b) Window Trace Function

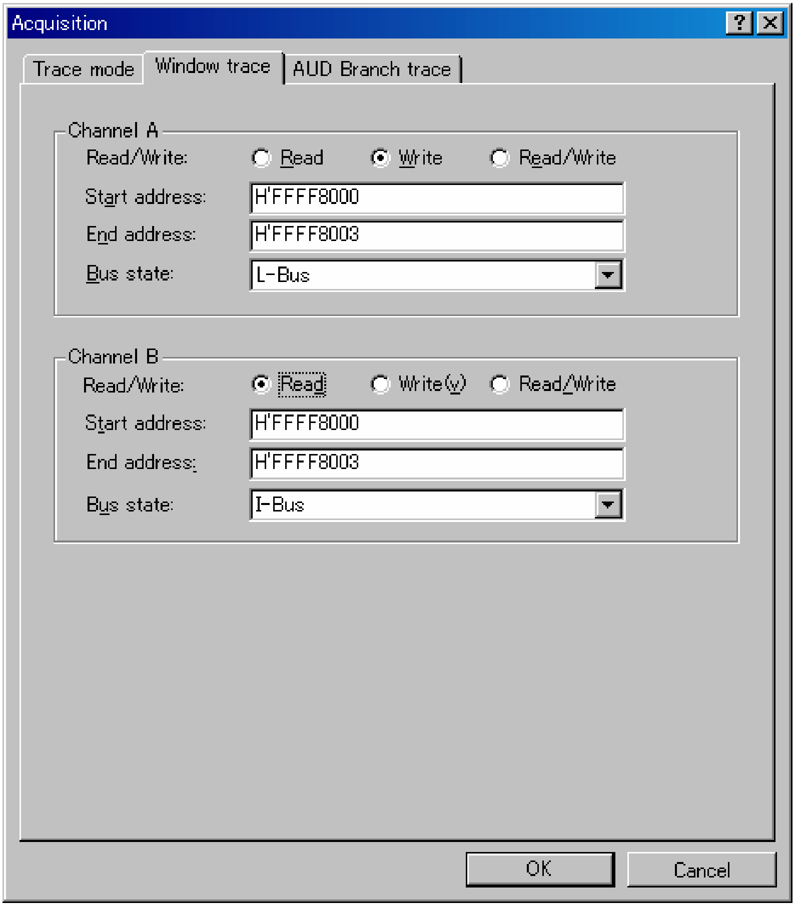
Memory access in the specified range can be acquired by trace.

Two memory ranges can be specified for channels A and B. The read, write, or read/write cycle can be selected as the bus cycle for trace acquisition.

[Setting Method]

(i) Select the [Channel A] and [Channel B] check boxes in the [AUD function] group box of the [Trace mode] page. Each channel will become valid.

(ii) Open the [Window trace] page and specify the bus cycle, memory range, and bus type that are to be set for each channel.



Channel A
Read/Write: ○ Read ● Write ○ Read/Write
Start address: H'FFFF8000
End address: H'FFFF8003
Bus state: L-Bus

Channel B
Read/Write: ● Read ○ Write(V) ○ Read/Write
Start address: H'FFFF8000
End address: H'FFFF8003
Bus state: I-Bus

Figure 2.5 [Window trace] Page

(c) Software Trace Function

Note: This function can be supported with SHC/C++ compiler (manufactured by Renesas Technology Corp.; including OEM and bundle products) V7.0 or later.

When a specific instruction is executed, the PC value at execution and the contents of one general register are acquired by trace. Describe the Trace(x) function (x is a variable name) to be compiled and linked beforehand. For details, refer to the SHC manual.

When the load module is downloaded on the emulator and is executed while a software trace function is valid, the PC value that has executed the Trace(x) function, the general register value for x, and the source lines are displayed.

To activate the software trace function, select the [Software trace] check box in the [AUD function] group box of the [Trace mode] page.

Notes on AUD Trace:

1. When the trace display is performed during user program execution, the mnemonics, operands, or sources are not displayed.
2. The AUD branch trace function outputs the differences between newly output branch source addresses and previously output branch source addresses. The window trace function outputs the differences between newly output addresses and previously output addresses. If the previous branch source address is the same as the upper 16 bits, the lower 16 bits are output. If it matches the upper 24 bits, the lower 8 bits are output. If it matches the upper 28 bits, the lower 4 bits are output.
   The emulator regenerates the 32-bit address from these differences and displays it in the [Trace] window. If the emulator cannot display the 32-bit address, it displays the difference from the previously displayed 32-bit address.
3. If the 32-bit address cannot be displayed, the source line is not displayed.
4. If a completion-type exception occurs during exception branch acquisition, the next address to the address in which an exception occurs is acquired.
5. The AUD trace is disabled while the profiling function is used.
6. Set the AUD clock (AUDCK) frequency to 32 MHz or lower. If the frequency is higher than 32 MHz, the emulator will not operate normally.
7. The AUD trace is disabled while the profiling function is used

2.2.4 Note on Using the JTAG (H-UDI) Clock (TCK)

1. Set the JTAG clock (TCK) frequency as a value that is lower than the frequency of the peripheral clock (Pϕ) and 1/4 or lower than the frequency of the bus clock (Bϕ).
2. The initial value of the JTAG clock (TCK) is 2.5 MHz.
3. A value to be set for the JTAG clock (TCK) is initialized after executing [Reset CPU] or [Reset Go]. Thus the TCK value will be 2.5 MHz.

2.2.5 Notes on Setting the [Breakpoint] Dialog Box

1. When an odd address is set, the next lowest even address is used.
2. A BREAKPOINT is accomplished by replacing instructions of the specified address. Accordingly, it can be set only to the internal RAM area.
   A BREAKPOINT cannot be set to the following addresses:
   — An area other than CS, the internal RAM, and the internal flash memory
   — An instruction in which Break Condition 2 is satisfied
   — A slot instruction of a delayed branch instruction
3. During step operation, specifying BREAKPOINTs and Event Condition breaks are disabled.
4. When execution resumes from the address where a BREAKPOINT is specified and a break occurs before Event Condition execution, single-step operation is performed at the address before execution resumes. Therefore, realtime operation cannot be performed.
5. If an address of a BREAKPOINT cannot be correctly set in the ROM, a mark ● will be displayed in the [BP] area of the address on the [Source] or [Disassembly] window by refreshing the [Memory] window, etc. after Go execution. However, no break will occur at this address. When the program halts with the event condition, the mark ● disappears.

2.2.6 Notes on Setting the [Event Condition] Dialog Box and the BREAKCONDITION_ SET Command

1. When [Go to cursor], [Step In], [Step Over], or [Step Out] is selected, the settings of Event Condition 3 are disabled.
2. When an Event Condition is satisfied, emulation may stop after two or more instructions have been executed.

2.2.7 Performance Measurement Function

The emulator supports the performance measurement function.

1. Setting the performance measurement conditions

   To set the performance measurement conditions, use the [Performance Analysis] dialog box and the PERFORMANCE_SET command. When any line in the [Performance Analysis] window is clicked with the right mouse button, a popup menu is displayed and the [Performance Analysis] dialog box can be displayed by selecting [Setting].

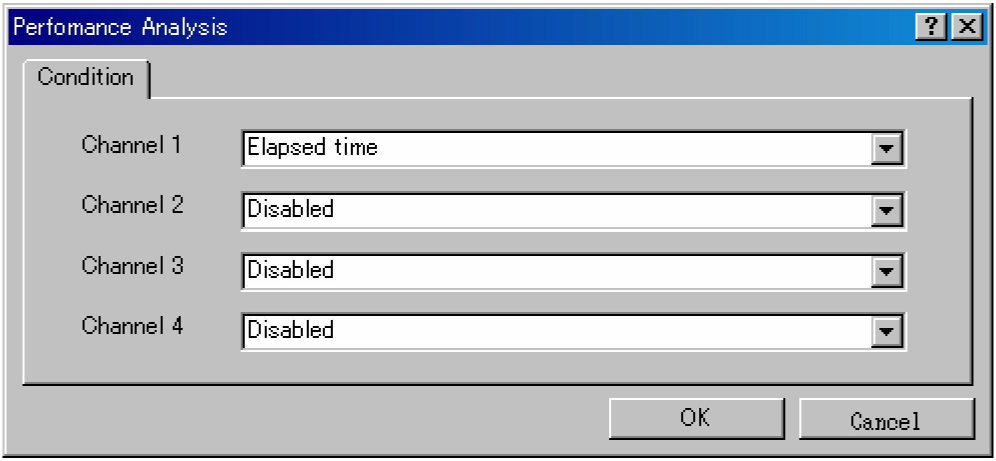
Note: For the command line syntax, refer to the online help.

(a) Specifying the measurement start/end conditions

The measurement start/end conditions are specified by using Event Condition 1,2. The [Ch1,2,3] list box of the [Combination action (Sequential PtoP)] dialog box can be used.

Table 2.14 Measurement Period

Classification	Item	Description
Selection in the [Ch1, 2, 3] list box	Ch2 to Ch1 PA	The period from the satisfaction of the condition set in Event Condition 2 (start condition) to the satisfaction of the condition set in Event Condition 1 (end condition) is set as the performance measurement period.
	Ch1 to Ch2 PA	The period from the satisfaction of the condition set in Event Condition 1 (start condition) to the satisfaction of the condition set in Event Condition 2 (end condition) is set as the performance measurement period.
	Other than above	The period from the start of execution of the user program to the occurrence of a break is measured.



Channel 1: Elapsed time

Channel 2: Disabled

Channel 3: Disabled

Channel 4: Disabled

Figure 2.6 [Performance Analysis] Dialog Box

For measurement tolerance,

- The measured value includes tolerance.
- Tolerance will be generated before or after a break.

Note: When [Ch2 to Ch1 PA] or [Ch1 to Ch2 PA] is selected, to execute the user program, specify conditions set in Event Condition 2 and Event Condition 1 and one or more items for performance measurement.

(b) Measurement item

Items are measured with [Channel 1 to 4] in the [Performance Analysis] dialog box. Maximum four conditions can be specified at the same time. Table 2.15 shows the measurement items.

Table 2.15 Measurement Item

Selected Name	Option
Disabled	None
Elapsed time	AC (The number of execution cycles (Iϕ) is set as the measurement item.)
Number of execution states	VS
Branch instruction counts	BT
Number of execution instructions	I
Exception/interrupt counts	EA
Interrupt counts	INT
URAM area access counts	UN
URAM area instruction access counts	UIN
URAM area data access counts	UDN

Note: Selected names are displayed for CONDITION in the [Performance Analysis] window. Options are parameters for <mode> of the PERFORMANCE_SET command.

Each measurement condition is also counted when a condition in table 2.16 is generated.

Table 2.16 Performance Measurement Condition to be Counted

Measurement Condition	Notes
Branch count	The counter value is incremented by 2. This means that two cycles are valid for one branch.

Notes: 1. In the non-realtime trace mode of the AUD trace, normal counting cannot be performed because the generation state of the stall or the execution cycle is changed.

2. When the CPU clock is halted in the mode, such as sleep, counting is also halted.

3. When the measurement start or end condition is set, counting is halted if a power-on reset is input after and before the satisfaction of measurement start and end conditions.

2. Displaying the measured result

The measured result is displayed in the [Performance Analysis] window or the PERFORMANCE_ANALYSIS command with hexadecimal (32 bits).

Note: If a performance counter overflows as a result of measurement, "********" will be displayed.

3. Initializing the measured result

   To initialize the measured result, select [Initialize] from the popup menu in the [Performance Analysis] window or specify INIT with the PERFORMANCE_ANALYSIS command.

SuperH™ Family E10A-USB Emulator Additional Document for User's Manual Supplementary Information on Using the SH7147 Group Debugging MCU Board

Publication Date: Rev.1.00, April 4, 2008
Published by: Sales Strategic Planning Div.
Renesas Technology Corp.
Edited by: Customer Support Department
Global Strategic Communication Div.
Renesas Solutions Corp.

Renesas Technology Corp. Sales Strategic Planning Div. Nippon Bldg., 2-6-2, Ohte-machi, Chiyoda-ku, Tokyo 100-0004, Japan

RENESAS SALES OFFICES

http://www.renesas.com

Refer to "**http://www.renesas.com/en/network**" for the latest and detailed information.

Renesas Technology America, Inc.
450 Holger Way, San Jose, CA 95134-1368, U.S.A
Tel: <1> (408) 382-7500, Fax: <1> (408) 382-7501

Renesas Technology Europe Limited
Dukes Meadow, Millboard Road, Bourne End, Buckinghamshire, SL8 5FH, U.K.
Tel: <44> (1628) 585-100, Fax: <44> (1628) 585-900

Renesas Technology (Shanghai) Co., Ltd.
Unit 204, 205, AZIACenter, No.1233 Lujiazui Ring Rd, Pudong District, Shanghai, China 200120
Tel: <86> (21) 5877-1818, Fax: <86> (21) 6887-7858/7898

Renesas Technology Hong Kong Ltd.
7th Floor, North Tower, World Finance Centre, Harbour City, Canton Road, Tsimshatsui, Kowloon, Hong Kong
Tel: <852> 2265-6688, Fax: <852> 2377-3473

Renesas Technology Taiwan Co., Ltd.
10th Floor, No.99, Fushing North Road, Taipei, Taiwan
Tel: <886> (2) 2715-2888, Fax: <886> (2) 3518-3399

Renesas Technology Singapore Pte. Ltd.
1 Harbour Front Avenue, #06-10, Keppel Bay Tower, Singapore 098632
Tel: <65> 6213-0200, Fax: <65> 6278-8001

Renesas Technology Korea Co., Ltd.
Kukje Center Bldg. 18th Fl., 191, 2-ka, Hangang-ro, Yongsan-ku, Seoul 140-702, Korea
Tel: <82> (2) 796-3115, Fax: <82> (2) 796-2145

Renesas Technology Malaysia Sdn. Bhd
Unit 906, Block B, Menara Amcorp, Amcorp Trade Centre, No.18, Jln Persiaran Barat, 46050 Petaling Jaya, Selangor Darul Ehsan, Malaysia
Tel: <603> 7955-9390, Fax: <603> 7955-9510

Colophon 6.2

SuperH™ Family E10A-USB Emulator Additional Document for User's Manual Supplementary Information on Using the SH7147 Group Debugging MCU Board

Renesas Technology Corp.
2-6-2, Ote-machi, Chiyoda-ku, Tokyo, 100-0004, Japan